

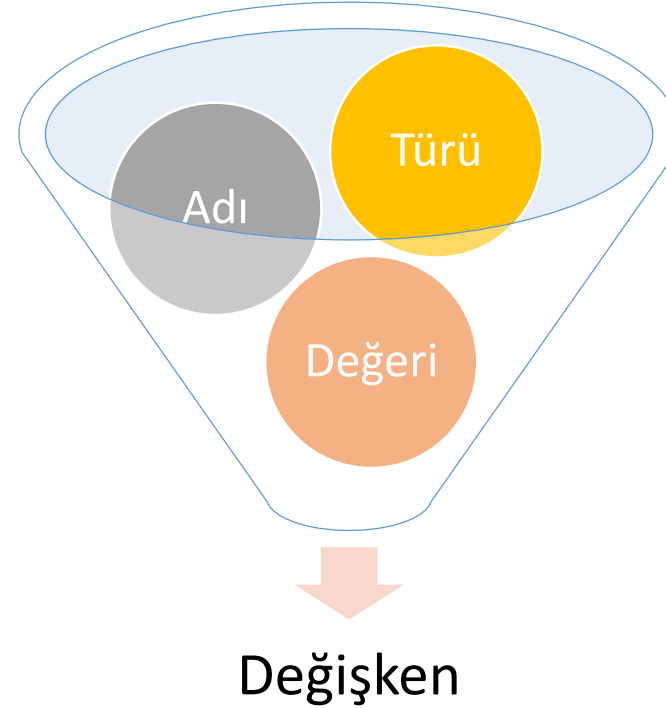
C# Sözdizilimini Öğrenme ve Temel Programlama Bileşenleri

Giriş

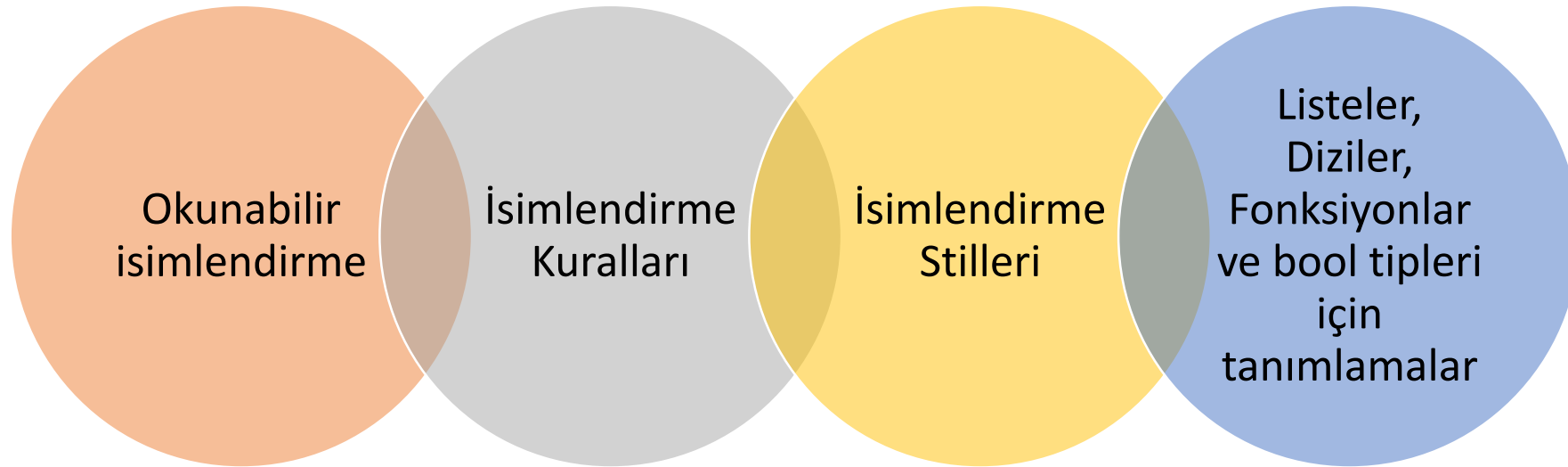
- Bu bölümde değişkenler, kontrol yapıları, çevrimler, diziler, listeler gibi temel programlama bileşenleri ile birlikte C# sözdizilimi konusunda bilgiler verilmektedir.

Değişkenler

- Hafızada bir yer ayırmak, gerekli durumlarda ayrılan yere değer atamak, değeri değiştirmek ve okumak için kullanılan programlama bileşeni değişken olarak adlandırılır.
- Bir başka ifadeyle değişken, bir değeri tutan depolama konumudur.
- Değişken, üç temel boyutu ile düşünülmelidir.
- Sabitler program boyunca değeri değişmeyen özel bir değişken türü olarak düşünülebilir.



Değişkenler

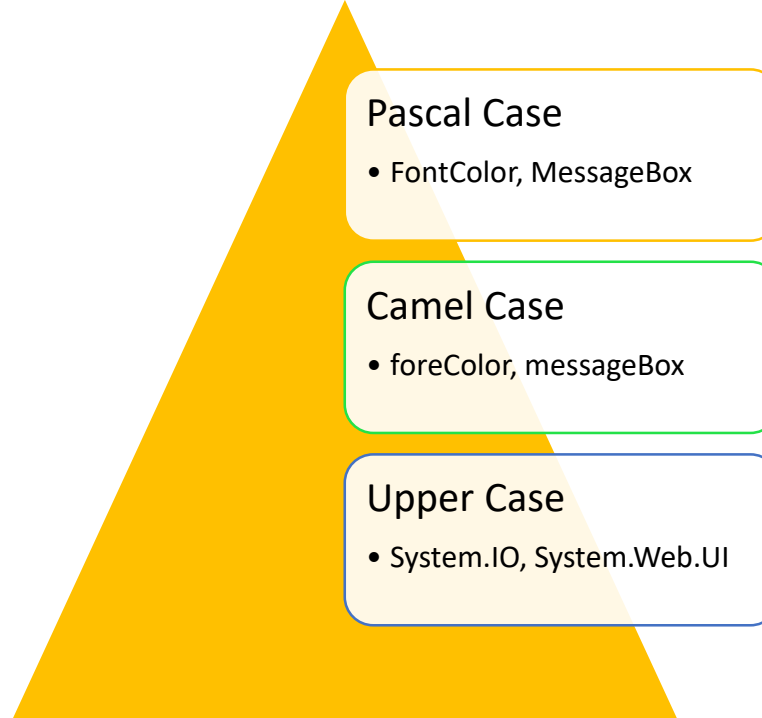


Değişken İsimlendirme

- Sürdürülebilir ve anlaşılabilir kodlar yazmak üzere isimlendirme için bir stil belirlemek önemlidir.
- Belirli bir isimlendirme stiline seçilmesi programlama sürecinde geliştiricilere sürdürülebilir kod yazma konusunda pozitif yönde katkıda bulunur.

Değişken İsimlendirme

- C# içerisinde metotları isimlendirirken, küçük ya da büyük harf ile metot isimlerini biçimlendirmede üç çeşit isimlendirme biçimi kullanılır.



Değişken İsimlendirme

- Sürdürülebilir ve anlaşılabilir, okunabilir kodlar yazmak üzere isimlendirme için bir stil belirlemek önemlidir.
- Belirli bir isimlendirme stiline seçilmesi programlama sürecinde geliştiricilere sürdürülebilir kod yazma konusunda pozitif yönde katkıda bulunur.

Değer Tipleri

Tipi	CTS Tipi	Kapladığı Alan
byte	System.Byte	8 bit işaretli tamsayı
sbyte	System.Byte	8 bit işaretli tamsayı
short	System.Int16	16 bit işaretli tamsayı
ushort	System.Int16	16 bit işaretli tamsayı
int	System.Int32	32 bit işaretli tamsayı
uint	System.UInt32	32 bit işaretli tamsayı
long	System.UInt64	64 bit işaretli tamsayı
ulong	System.UInt64	64 bit işaretli tamsayı

Değer Tipleri

Tipi	CTS Tipi	Kapladığı Alan
float	System.Single	32 bit tek kayan sayı
double	System.Double	64 bit çift kayan sayı
decimal	System.Decimal	128 bit ondalıklı sayı
bool	System.Boolean	
char	System.Char	16 bit Unicode

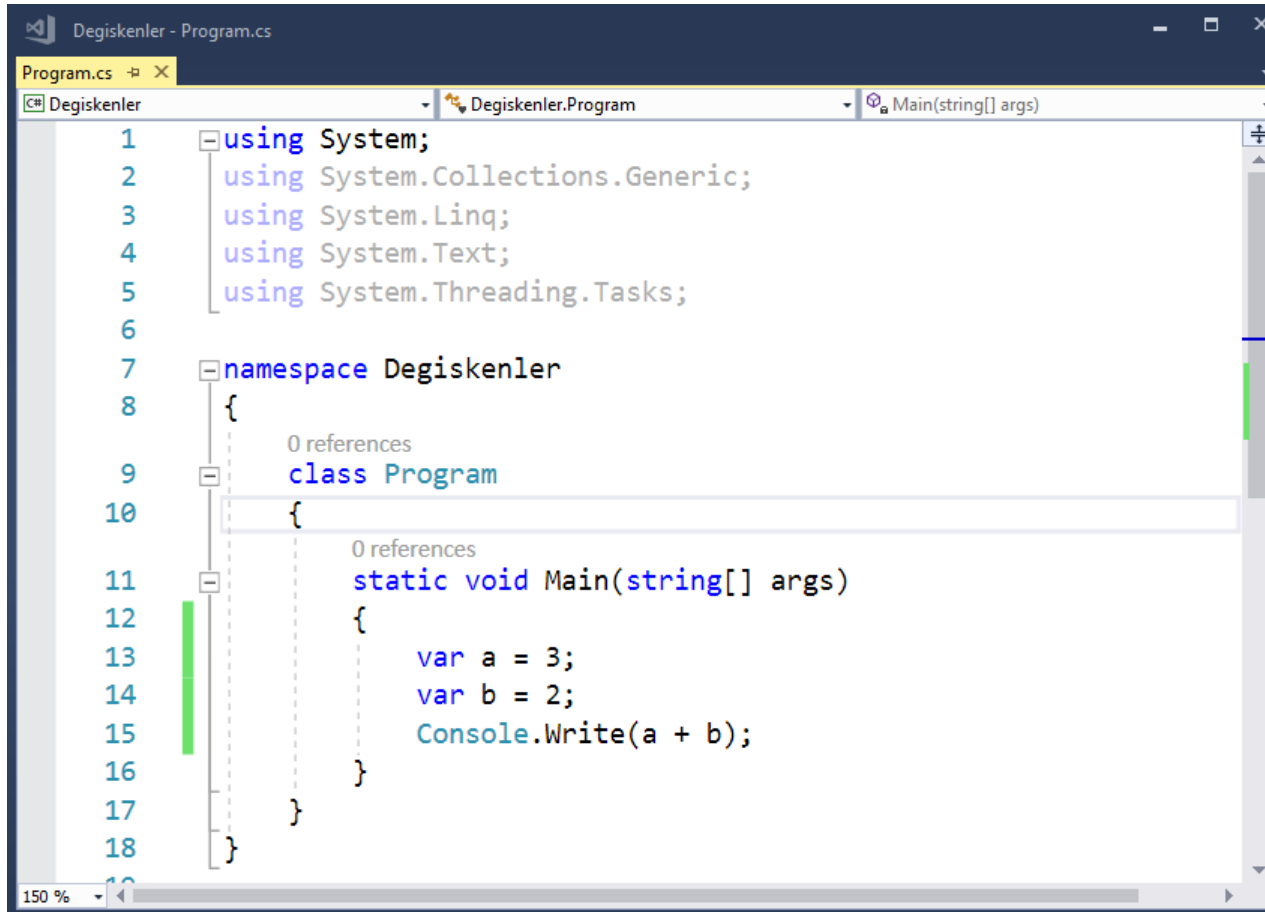
Referans Tipleri

Değişken Tipi	Açıklama
string	Unicode karakterlerinden oluşan string.
object	Genel veri tipi. Tip dönüşümlerinde oluşan hatalar derleme esnasında göz ardı edilerek çalışma zamanında hata üretilir.
dynamic	Genel veri tipi. Tip dönüşümlerinde oluşan hatalar üzerinde derleme esnasında herhangi bir kontrol yapılmadığı için hata üretilmez.

var

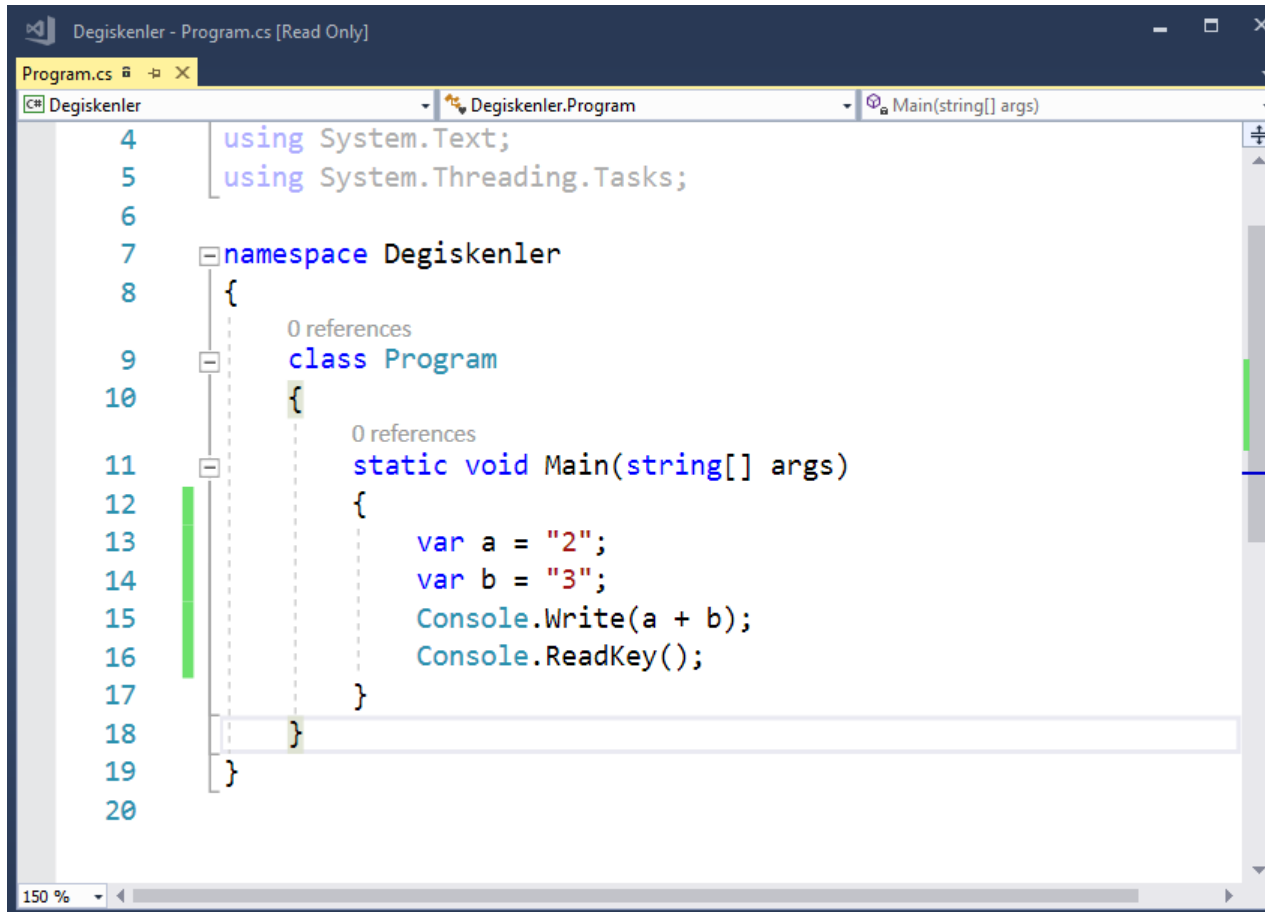
- Değişken tanımı yapılırken tip belirtilmeden **var** deyimi kullanılabilir.
- **var** ifadesi ile tanımlanan değişkenler için ilk değer ataması yapılmalıdır.
- Değişkenin tipi kullanılan veri türüne bağlı olarak C# tarafından otomatik olarak belirlenecektir.

var



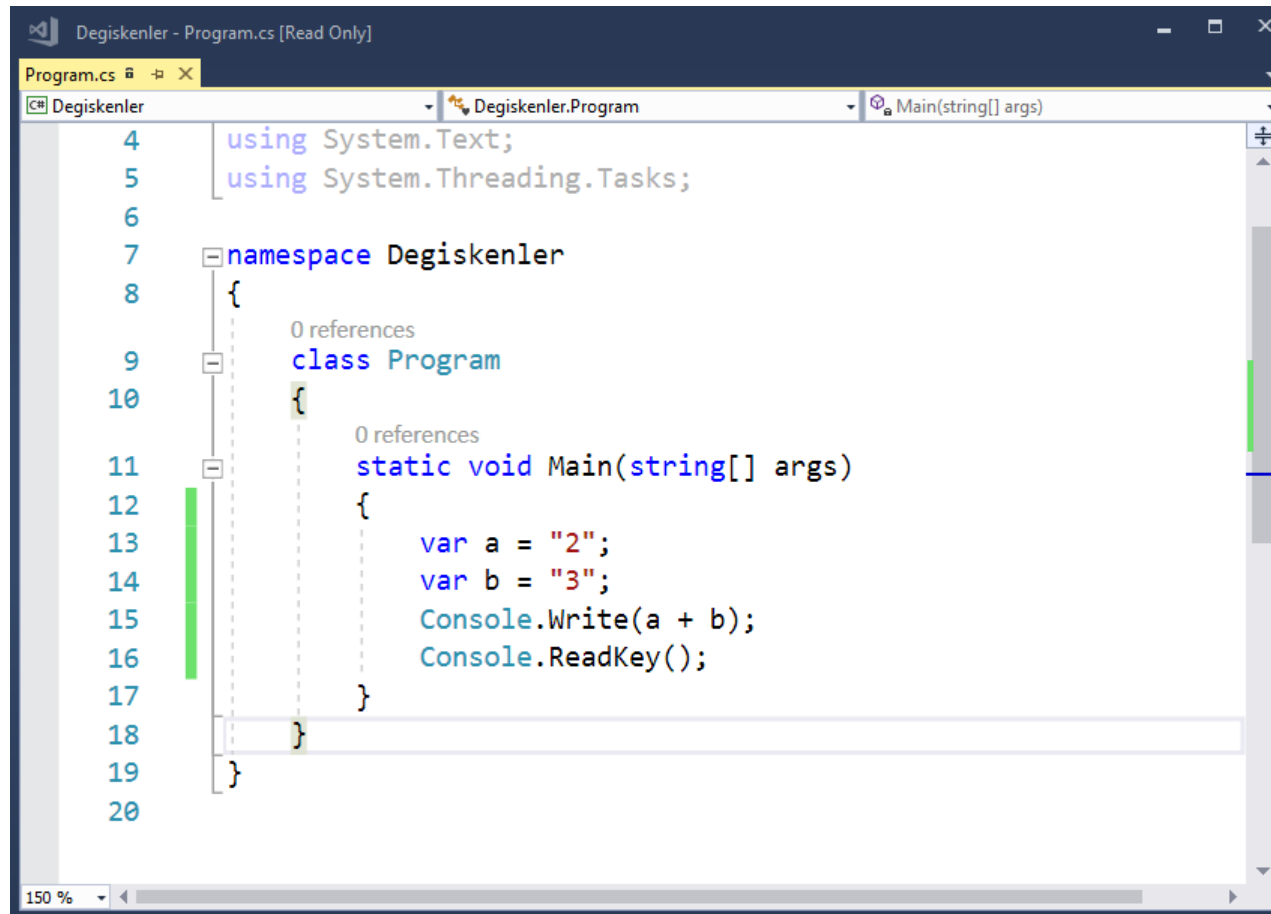
```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace Degiskenler
8  {
9      class Program
10     {
11         static void Main(string[] args)
12         {
13             var a = 3;
14             var b = 2;
15             Console.WriteLine(a + b);
16         }
17     }
18 }
```

var



```
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace Degiskenler
8  {
9      0 references
10     class Program
11     {
12         0 references
13         static void Main(string[] args)
14         {
15             var a = "2";
16             var b = "3";
17             Console.Write(a + b);
18             Console.ReadKey();
19         }
20     }
```

Digit Seperator

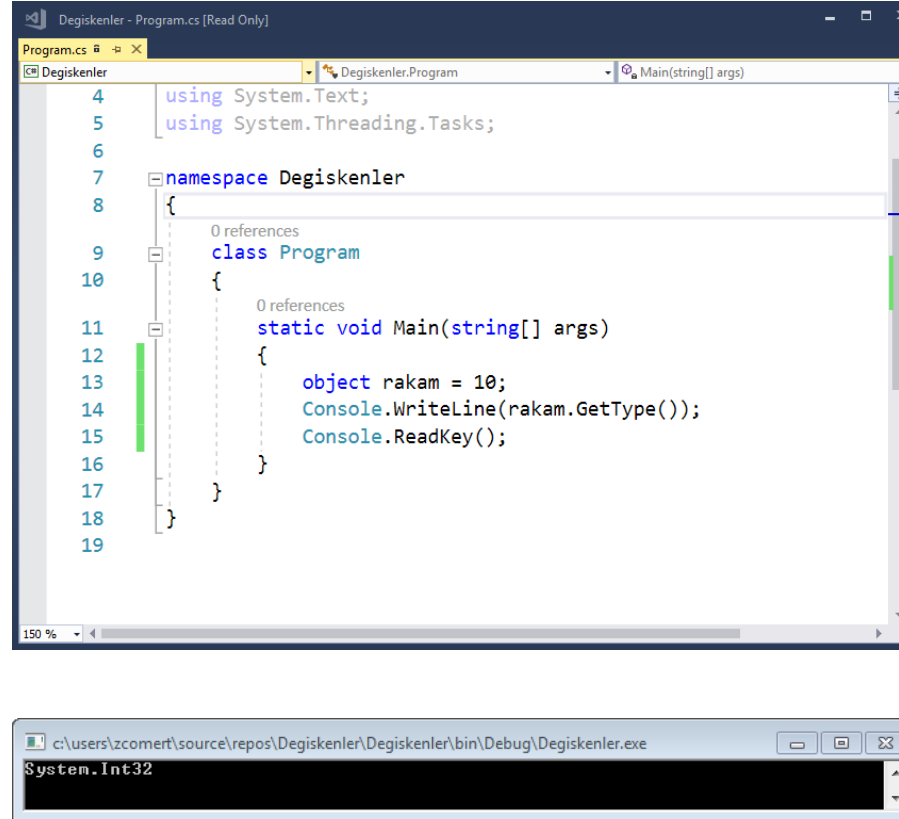


```
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace Degiskenler
8 {
9     class Program
10     {
11         static void Main(string[] args)
12         {
13             var a = "2";
14             var b = "3";
15             Console.Write(a + b);
16             Console.ReadKey();
17         }
18     }
19 }
20
```

C# 7 versiyonu ile gelen **Digit Seperator** özelliği ile rakamları basamaklarına ayırarak daha düzenli bir şekilde yazabilirsiniz.

GetType

- Kullanılan ya da tanımlanan veri türünün tipi almak üzere **GetType** deyimi kullanılmaktadır.



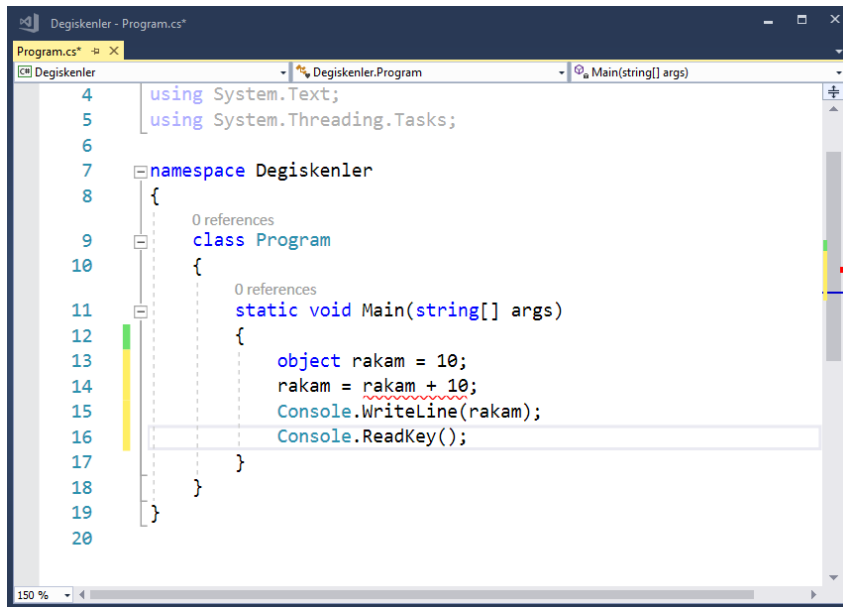
The image shows a Visual Studio code editor window titled "Degiskenler - Program.cs [Read Only]". The code is as follows:

```
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace Degiskenler
8 {
9     0 references
10    class Program
11    {
12        0 references
13        static void Main(string[] args)
14        {
15            object rakam = 10;
16            Console.WriteLine(rakam.GetType());
17            Console.ReadKey();
18        }
19    }
```

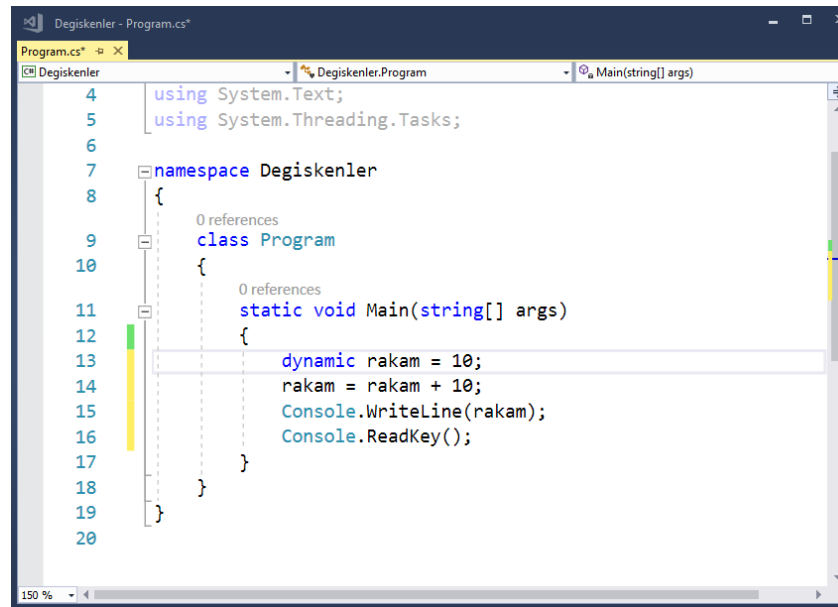
Below the code editor, a console window titled "c:\users\zcomert\source\repos\Degiskenler\Degiskenler\bin\Debug\Degiskenler.exe" shows the output: "System.Int32".

object ve dynamic anahtar sözcükleri

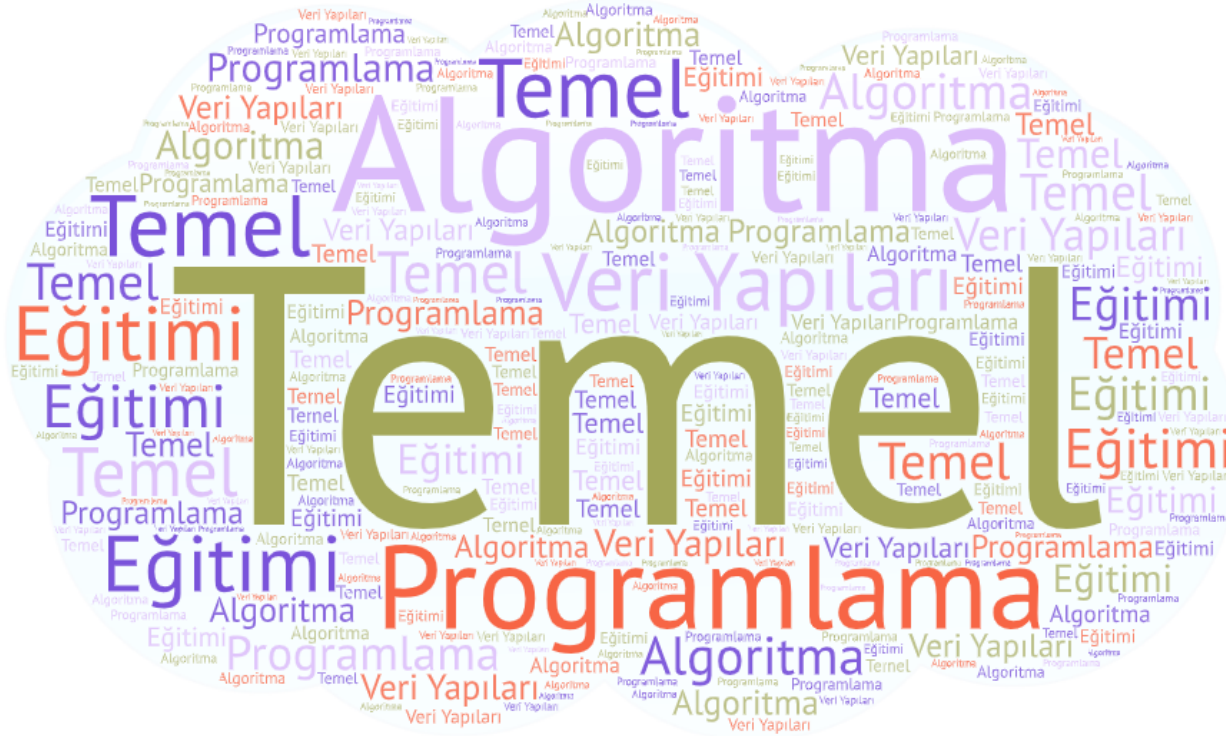
- Operator '+' cannot be applied...



```
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace Degiskenler
8 {
9     0 references
10     class Program
11     {
12         0 references
13         static void Main(string[] args)
14         {
15             object rakam = 10;
16             rakam = rakam + 10;
17             Console.WriteLine(rakam);
18             Console.ReadKey();
19         }
20     }
```



```
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace Degiskenler
8 {
9     0 references
10     class Program
11     {
12         0 references
13         static void Main(string[] args)
14         {
15             dynamic rakam = 10;
16             rakam = rakam + 10;
17             Console.WriteLine(rakam);
18             Console.ReadKey();
19         }
20     }
```

C# Sözdizilimini Öğrenme ve Temel Programlama Bileşenleri